



# Calculating Positioning Error in the Flying Wires Beam Profiling System

Joshua Hooks

University of Maryland Baltimore County

Fermi National Accelerator Laboratory

SIST Program 2010

Supervisor: Jim Zagel

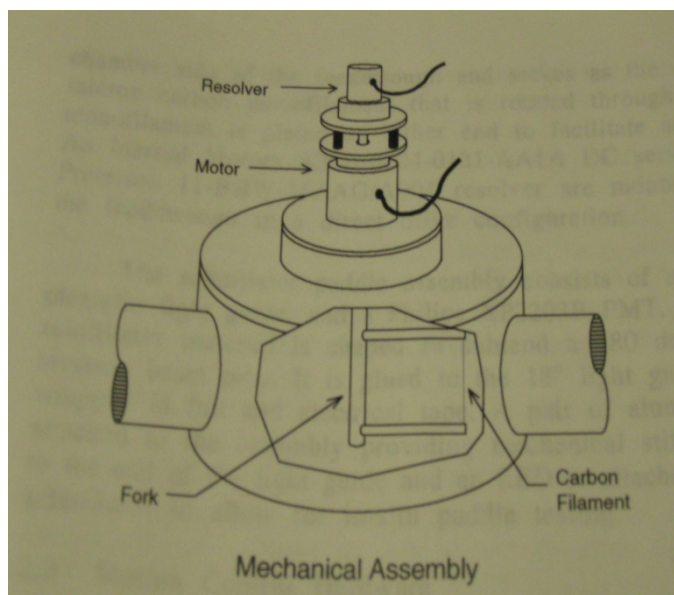
August 3, 2010

## **Abstract**

The ability to accurately profile a proton and antiproton beam is necessary for further research into their collisions at Fermi National Accelerator Laboratory. To achieve these accurate results, the position of the wire, used in the Flying Wire beam profiling system, must be known as precisely as possible. In this paper, a method for finding the amount of error in the recorded position of the wire is explored and analyzed. A program, created using National Instruments LabVIEW [3], compares position readings from a resolver to the actual positions given by an optical encoder. After testing, we found that the position as reported by the resolver is typically within 10 counts of the actual position given by the optical encoder. Using the program developed this summer, the effects of speed, coupling type and other design parameters can be tested before being used in the accelerator tunnels for their overall affect on measurement accuracy.

## Introduction

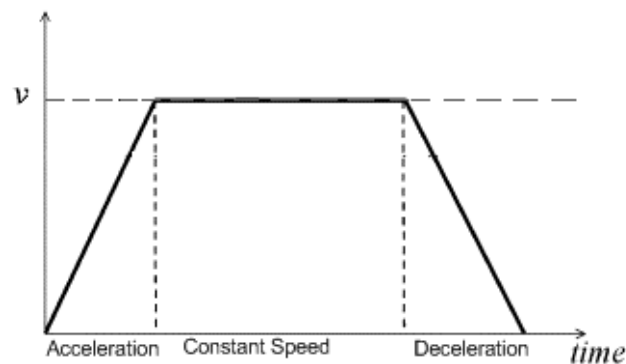
Fermi National Accelerator Lab is a government funded High-Energy Physics Laboratory. This lab studies subatomic particles by accelerating protons and antiprotons to near light speeds and then colliding them into each other. Through these collisions, even more basic particles are released and can be further studied. The proton and antiproton beams are manipulated in specially designed tunnels, such as the Main Injector and Tevatron, which use magnets and electrical currents to guide and accelerate the two beams. While in these tunnels, detectors, known as flying wires [2], are used to profile the proton and anti proton beams. They do so by spinning a wire attached to two ends of a fork through the beams. A device known as a resolver is used to locate the absolute rotational position of the wire at any time during a revolution. The position of the wire as given by the resolver can be slightly inaccurate, while the wire is in motion. The goal of this project was to find this error and the factors affecting this error, by comparing the position given by the resolver to the position given by an optical encoder mounted on the output shaft of the driven fork.



**Figure 1: The schematics of the flying wire beam profiling system.**

The flying wire system measures the transverse size, a cross-section of the beam, by passing a 33 micron to 7 micrometer carbon filament (wire) through the beam [2]. The “flying wire” has been drawn tight between the prongs of a metal fork, and the fork is attached to a

motor. The resolver, which is mounted above the motor (see figure 1), has reports the position of the fork as it is driven through a 540 degree revolution in a manner that will allow the wire to pass through the beam twice during each run [1]. Passing through the beam twice allows differences to be averaged and more accurate results. During the spin, the fork must accelerate so that the wire will be moving at a constant 6 m/s when passing through the beam twice and decelerate in time so that the system can do the exact same spin in the opposite direction [2]. Figure 2 shows the three different aspects of the motion profile. When the wire passes through the beam some protons will hit the wire creating a shower of secondary particles. These particles that are created are picked up by large scintillator paddles with photo multiplier tubes [1], [2]. After the wire has made a complete pass, the profile and intensity of the beam can be calculated based on the number of particles and their measured positions [2].



**Figure 2: A typical trapezoidal velocity graph through one revolution of the flying wire system. Note the distinct acceleration, constant speed and deceleration portions of the motion.**

In order to constantly know the position of the wire and to ensure consistent results, a resolver has been installed for motion control in the tunnels, replacing an optical encoder. The optical encoder was sensitive to the radiation in the tunnels. The resolver uses two coils that emit signals replicating the phases of sine and cosine waves [4]. Using these coils allow the resolver to constantly give the position of the wire in the tunnel. This position is given in reference to an “absolute zero” or index position in the rotation, and is known as the absolute position [2]. Due to this method in which position is found, a slight, but noticeable, lag time occurs during rotation, causing error in the position readings. The position is first sent to the Resolver Interface

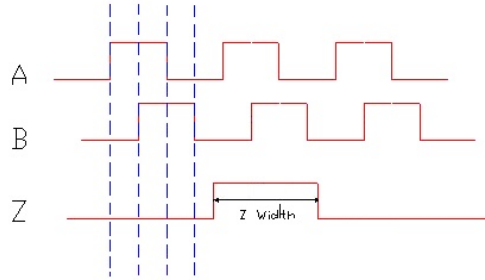
Position FIFO (first in, first out) module, commonly known as the “RIPFIFO”, which provides an industry standard encoder signal to the motion control system.

The optical encoder gives position relative to an index mark. Figure 3 illustrates the three different tracks on an optical encoder; A, B, and Z (also known as the index). Both the A and B track are composed of 4,096 marks each. Every time a space between a mark occurs, an L-E-D (Light-Emitting Diode) light shines through creating a signal. The same occurs on the Z track, but there is only one index space. All following position values of a rotating body can only be put in terms relative to what the index value was and when the value was last received.



**Figure 3:** This shows the inside disk of an optical encoder, with the A, B, and Z tracks. The LED light shines through the spaces to make the signals that we can then view on an oscilloscope display.

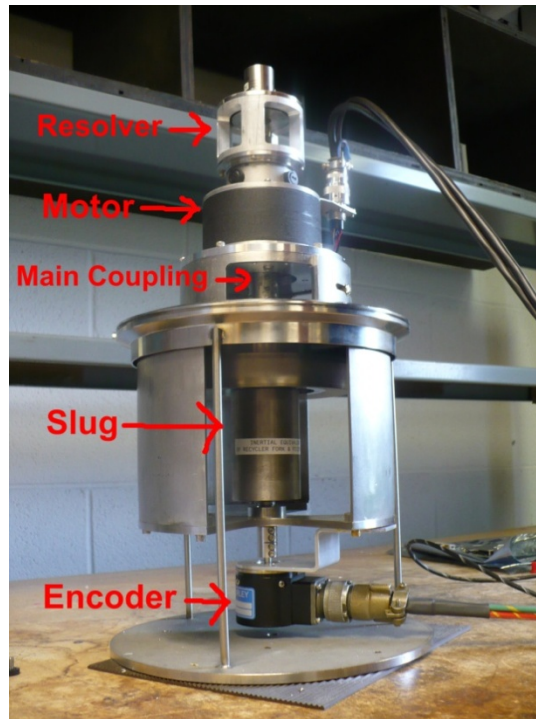
The optical encoder is read with an oscilloscope. This machine displays the occurrence of these marks and spaces as a function of time. A high section on the display means that the LED light is shining through a space and low area of a trace signifies a blocked portion. The A phase and B phase are offset by 90 degrees, therefore when comparing a period of the two phases on an oscilloscope we see 4 unique events: A high-B low, A high-B high, A low-B high, and A low-B low, see Figure 4. By using these 4 states of the two periods, the accuracy in recording position is quadrupled. This method is known as quadrature decoding. The position of a rotating device is given in counts where a count is the duration of one of these events (i.e. A high- B high). Since there are 4,096 spaces for each track, by using quadrature we get that every rotation has 16,384 counts.



**Figure 4: This is how the optical encoder reading is displayed by an oscilloscope. Notice that the single index pulse is larger than a phase A or B pulses.**

### **Materials and Methods**

National Instruments LabVIEW programming was used during the duration of the research to create a program that could control the motion of a flying wire model, and then compare the positions, at specific points in time, from the optical encoder and the resolver. An external oscilloscope was used to find the rate of position changes that occurred after the index pulse had passed. At this point, the recorded encoder positions at certain time increments could be compared to the positions that the resolver gave at the same times. Based off these comparisons, the error of the resolver recorded position could be compared at different velocities, with different couplings, and by varying other design parameters.



**Figure 5:** Above is the flying wire model used to calculate position error. The biggest difference is that the fork has been replaced with a slug for simplicity. In addition, an optical encoder has been mounted on the bottom.

A model of the flying wire system was made that would be better suited for comparing positions/calculating error. The set up of this model was simple, and mostly the same as the actual flying wire device. Figure 1 and 5 can be compared to see the difference in the position error model to the standard. On the very top of the model was a resolver, which provides position information to the RIPFIFO motion control system connected to the computer. This information is also used to drive the motor to a given position relative to an absolute zero that is programmed. The motor, controlled by the resolver, is mounted above the slug by a coupling. The slug for this model replaces what would be a feed through and fork in the actual tunnel. The motor drives the slug to the programmed position. The slug's spinning rotates the A, B, and Z tracks on the optical encoder that is mounted under it by a standard solid coupling. By designing the model in this manner, the resolver can drive the slug to a user specified position while the encoder reads the change in position that the slug is actually undergoing.

The first step toward finding error in this project was learning to get position readings from the encoder. The encoders can be read using an oscilloscope to record the encoder waveforms

produced as explained in the introduction. For our purposes we only want a portion of that waveform graph during constant velocity, acceleration, or deceleration. In order to get a single time frame on an oscilloscope a trigger is used. This trigger is usually a single pulse from some source that when detected, a timeframe of pre-trigger and/or post-trigger information can be collected. The index pulse acts as a trigger, after which we can collect phase A and B waveforms for a set amount of time.

A first attempt to read the encoder using onboard scope cards proved to be unsuccessful. An onboard scope card essentially acts as an oscilloscope built into the computer. When connected to the flying wire model, errors with memory depth and general processing caused the waveform graphs produced to be warped and shifted. For this reason, the encoder was eventually set up to an external scope. A LabVIEW program was created so that the external triggers settings and data could be set and read from the computer. We captured roughly 16 milliseconds of waveform after the index hit. As stated earlier, the encoder's position is all relative, and has no "knowledge" of the absolute position the resolver is recording and that is used as an input. To solve this problem, we used an Arbitrary Waveform Generator. When triggered, this device can release a programmable number of pulses with a given period and width. The waveform generator was set to be triggered by the index pulse along with the scope. Using LabVIEW, the number of rising edges of Phase A and B can be found between two generated pulses. Consequently, the position change, in counts, in a given time frame is found from the encoder. The number of rising and falling edges corresponds to the counts the encoder measures that the slug has rotated through. Since the position change of the slug after the index is known, the only problem left to overcome is to find the location of the index in relation to the absolute position that the resolver measures.

The waveform generator has a dual purpose. Along with allowing us to find position change in given time intervals for the encoder, the pulses created can trigger the RIPFIFO board to read the current resolver position. This way, for every pulse generated, we get a corresponding position value. Since the first pulse is generated when the index hits, the corresponding position from the RIPFIFO allows us to know where the index is in terms of the absolute positioning of the resolver. By adding the values of change in position from the encoder to this initial index position we can now compose a list of positions from the encoder and their corresponding



position from the resolver.

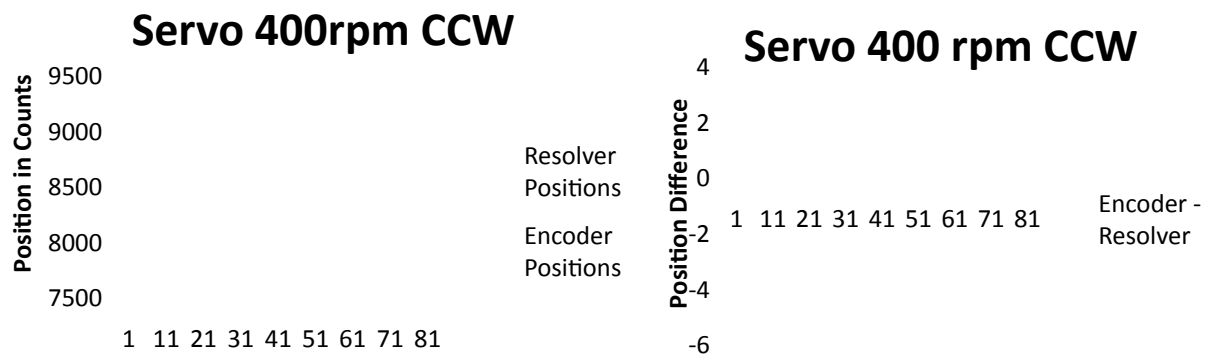
Now that the encoder and resolver have been synchronized to give back comparable position readings, testing can begin. A primary function of the program is to test how velocity and acceleration affect the position comparisons. To do this, the index was placed in a range that would allow us to test the different motion portions of a rotation. In order to compare positions during constant velocity, the index must be placed as close to the middle of our spin as possible. Therefore bounds were placed 8192 counts above and below the found index position for a  $360^\circ$  revolution. For a  $540^\circ$  revolution the bounds were placed 12288 counts above and below the index. This ensures that the index will be “hit” in the middle, and most likely constant speed portion, of each spin. With this setting the slug should have the maximum time possible to accelerate and decelerate without those changes being reflected in our 16 millisecond capture. It is important to note that the first run after switching from  $360^\circ$  to  $540^\circ$  (or vice versa) on the front panel will cause a partial spin or an elongated spin (for example: from the lower bound of a  $360^\circ$  to the upper bound of a  $540^\circ$  revolution). Therefore one unrecorded spin should be allowed when degree lever is flipped on the front panel.

In order to place the index in the accelerating or decelerating portion of the revolution proved to be more complex than placing the index in constant speed. First, the number of counts the system would be accelerating for was determined. Since the acceleration and deceleration are always 100 RPS/s, the factor affecting time was the desired top speed. A higher top speed allowed a greater time frame for acceleration. After some calculations, it was found that the minimum top speed of 424 RPM (used in the main injector) would allow .07066 seconds during the acceleration portion and consequently about 4,089 counts. Therefore, the index should be made to be around 2040 counts from the lower bound to place the trigger safely in the accelerating portion of the spin. For the spins to be reproducible for multiple runs the upper bound should always be precisely 16384 counts larger than the lower bound, making all tests for acceleration and deceleration  $360^\circ$  spins. Consequently, all clockwise spins directly correlate to acceleration tests, and counterclockwise spins become deceleration tests. Again, it is important that for all these spins the target speed is made to be 424 RPM and that the  $360^\circ$  flip is switched.

Along with constant speed and acceleration, another design parameter tested for its effect on the error during motion was the coupling connecting the motor to the slug. A coupling is

simply a device used to connect two shafts together at their ends for the purpose of transmitting power/rotation. The first coupling was a Servometer brand flexible coupling (model SMC-57), abbreviated in this document as Servo. A flexible coupling transmits torque while allowing a mild amount of radial, axial, and angular misalignment. This means that even with some misalignment a flexible coupling still allows general “smooth” rotation. Also there can be some twisting of the coupling, allowing the top (attached to motor) to start spinning, before the lower part overcomes the inertia of the slug. Another type of coupling used is a solid/rigid coupling. The solid coupling used in this testing was custom made by Fermilab. This coupling is used when precise shaft alignment is required. The trade off is that any misalignment will more noticeably affect the couplings performance, typically causing “rougher” or more grinding spins. Both of these couplings were tested under constant speeds, from 400-650 revolutions per minute (in 50 RPM increments), As well as under accelerating and decelerating conditions. Each coupling had slightly different effects on error under all these conditions.

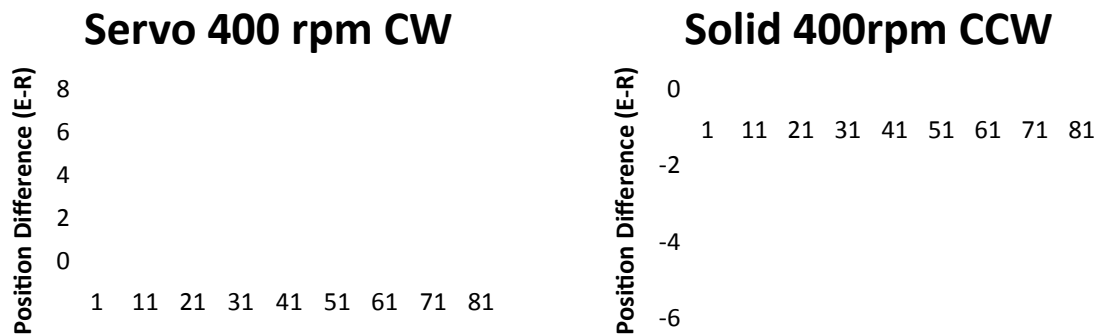
### Results/Discussion



**Figure 6: The graph above compares values of the Resolver and Encoder positions over the 16 millisecond capture and its' consequent difference in positions graph. The difference between the two position readings only vary by  $\pm 5$ , causing the encoder trace to mostly overlap the resolver trace in the position graph.**

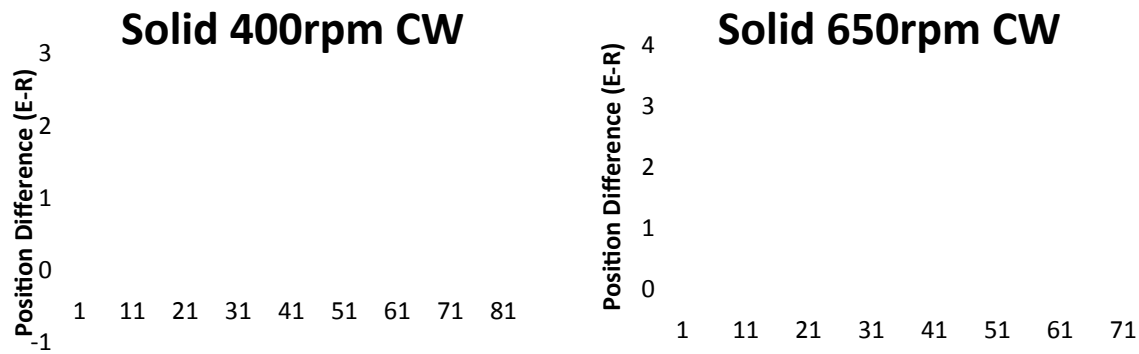
As expected, a position difference was found between the resolver and the optical encoder. This error is relatively small, with position differences typically ranging by  $\pm 6$  counts. Figure 6 provides an example of this. For all spins, the encoder seems to be “ahead of” the resolver in position readings. The encoder is reporting positions that the resolver won't report until a later time. This explains why clockwise spins report a positive difference in the Encoder-

Resolver positions, and why counter-clockwise spins typically have negative differences (see Figure 7). This, in part, can be explained by the process in which the two positions are read. Whereas the encoder position is immediately recorded, the resolver position must first go to the RPFIFO before being read by the computer. An additional lag is caused by the way the resolver is set up, with the sine and cosine coils. These explanations all contribute to why the encoder's position is a few counts in front of the resolver's. The only problem with accepting this explanation is that it does not explain why the position difference is increasing over time as seen in the graphs. In order to address this increasing error, inspection into top speed, acceleration, and deceleration were explored.



**Figure 7:** Notice that for a clockwise spin and counterclockwise spin, the encoder positions are a few counts in front of the resolver positions. To “be ahead” for a clockwise spins that means the position value is a larger number, and for counter-clockwise spins “being ahead” corresponds to being a smaller value.

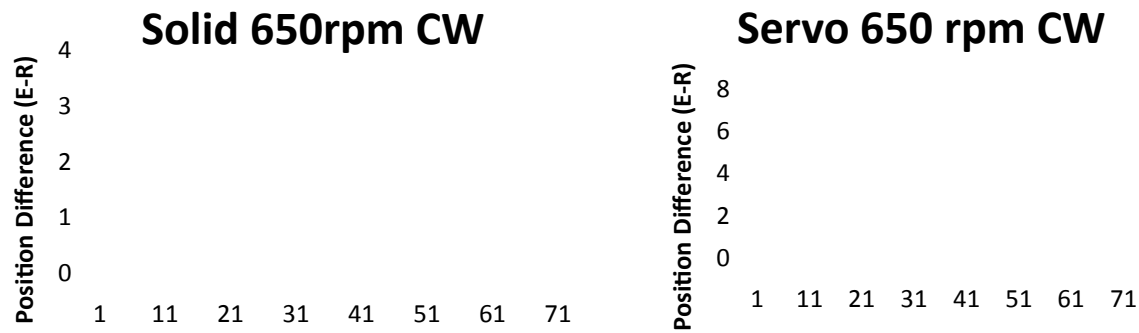
The top speed set by the program does not seem to make that big of a change in the difference between positions as reported by the encoder and resolver. As speed is increased from 400 RPM to 600 RPM, the extremes for difference only really increase by a count. For example, the Servometer coupling had a maximum difference in position (between the encoder and resolver) of about 5.6 counts for clockwise spins at 400 RPM. When this speed was increased to 650 RPM, the maximum difference between the encoder and resolver became about 6.5 counts. While noticeable, a one count difference is largely negligible. Top speed made little difference regardless of coupling type or spin direction. Increasing speed did seem to create “smoother” runs as seen in Figure 8. Smooth runs kept small oscillations between position differences from one point to the next to a minimum.



**Figure 8: Higher speeds caused for “smoother runs”. With less frequent oscillation in error from point to point.**

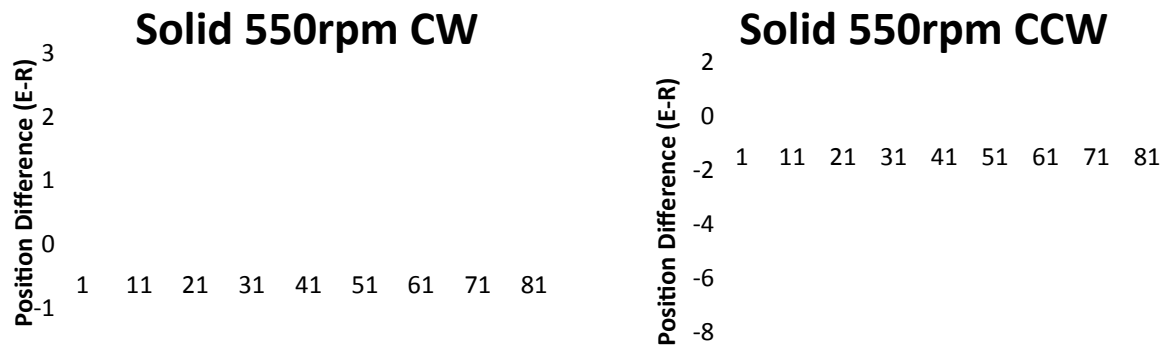
Changes in motion in general, such as top speed, acceleration, and deceleration did not seem to really change the amount of error accumulated. Placing the Index pulse in the acceleration or deceleration portion of the graphs did not seem to cause any great change. The magnitude of differences between the two positions were not noticeably greater or less when data was collected during acceleration/deceleration than under constant speed. Top speed and acceleration do not appear to be the lead factors affecting increasing error.

Changing the coupling had a significant effect on the magnitude of the difference between the positions. Particularly for clock wise spins, the solid coupling appears to have significantly less error. On average the maximum position difference for the solid coupling on clock wise spins was about 2.5 counts. On similar spins for the Servometer flexible coupling, error on average was around 6 counts. An example of this error difference based on coupling type is seen in Figure 9. The difference in position is more than doubled for the Servometer flexible coupling. Another important thing to note for the solid coupling is that the graphs appear to be more “jagged”, with the error between the resolver and encoder oscillating noticeably. This seems to imply that although the error is kept smaller with the solid coupling, the spin is less fluid. This does make sense given the nature and intentions of these couplings.



**Figure 9: The Solid Coupling and significantly less position difference (by about 3 counts) than the Servometer flexible coupling for clockwise spins.**

The factor that appears to cause the greatest error between the positions was the direction of the spin. Counter-clockwise spins increase the amount of error between the encoder and resolver. Although the sign changes in the position difference graphs are to be expected, the fact that the graphs completely change shape seem strange. Particularly for the solid coupling, counter-clockwise spins change the magnitude of position difference from 2.5 counts to near 8 counts as seen in Figure 10. Direction makes less of a error magnitude change for the flexible Servometer coupling, but still completely alters the shape of the graph. This seems to imply that there is significant misalignment with the current set up of the flying wire model that this testing was done on. This explains why the position difference for the solid coupling is so dependent on direction, because solid couplings do not compensate for misalignments as well as flexible couplings. Therefore, if there is some sort of snag that occurs when spinning counter-clockwise this will have a significant effect on the solid coupling's rotation. The difference in graph shape is most likely due to the fact that the 16 millisecond capture periods were taken during different sections of the spin. The index was placed exactly in the middle, so 16 milliseconds after the fly clockwise would not be the same section of the spin as 16 milliseconds after the index in the opposite direction.



**Figure 10: When spinning clock wise, the solid coupling kept the resolver and encoder within 3 counts of each other, but counter-clockwise spins increased this difference to over 6 counts.**

### Conclusion

The results from the tests on the flying wire model largely fit with what would be expected. Increasing the target speed of the system did lead to a mild increase in error, although in general the system was able to handle the increased stress caused by increasing top speed well. The coupling tests also were consistent with predicted expectations. The flexible coupling did have greater error in position, but was able to cope with design stresses better. Alternatively, the solid coupling was able to keep the position between the resolver and optical encoder closer in values, but was more susceptible to alignment errors. Using the program developed this summer, the effects of speed and coupling type can be tested. In the future all changes to the flying wire beam profiling detector can be tested using this model and program to ensure that errors do not accumulate and that profiling calculations are kept as accurate as possible.

## **Acknowledgments**

I would like to acknowledge my supervisor, James Zagel, for his guidance and dedication to my success while working on my project. I would like to thank my mentor Dave Peterson for all his advice and contributions to this project. I also must acknowledge Dave Slimmer, who gave me invaluable assistance and insight into LabVIEW programming. Special thanks go to the SIST program committee who provided me this opportunity to do research at Fermi National Accelerator Lab.

## References:

- [1] G. Tassotto and J. Zagel, "Beam Profile Detectors at the New Fermilab Injector and Associated Beamlines", Presented at the Proceedings DIPAC 1999, Chester, UK, 16-18 May 1999.
- [2] G. Vogel, W. Blokland, and J. Dey, "*NEW FLYING WIRE SYSTEM FOR THE TEVATRON*", Presented at the PARTICLE ACCELERATOR CONFERENCE, Vancouver, B.C., Canada 12-16, May, 1997.
- [3] J. Travis and J Kring, *LabVIEW for Everyone: Graphical Programming Made Easy and Fun*. Upper Saddle River, NJ: Pearson Education, Inc, 2007.
- [4] T. Nakamura, "Position Sensor," [Online document], 2008 Sept. 16 (Rev. 1.2.4.), [cited 2010 July. 17], Available: <http://www.faqs.org/patents/app/20100066347>